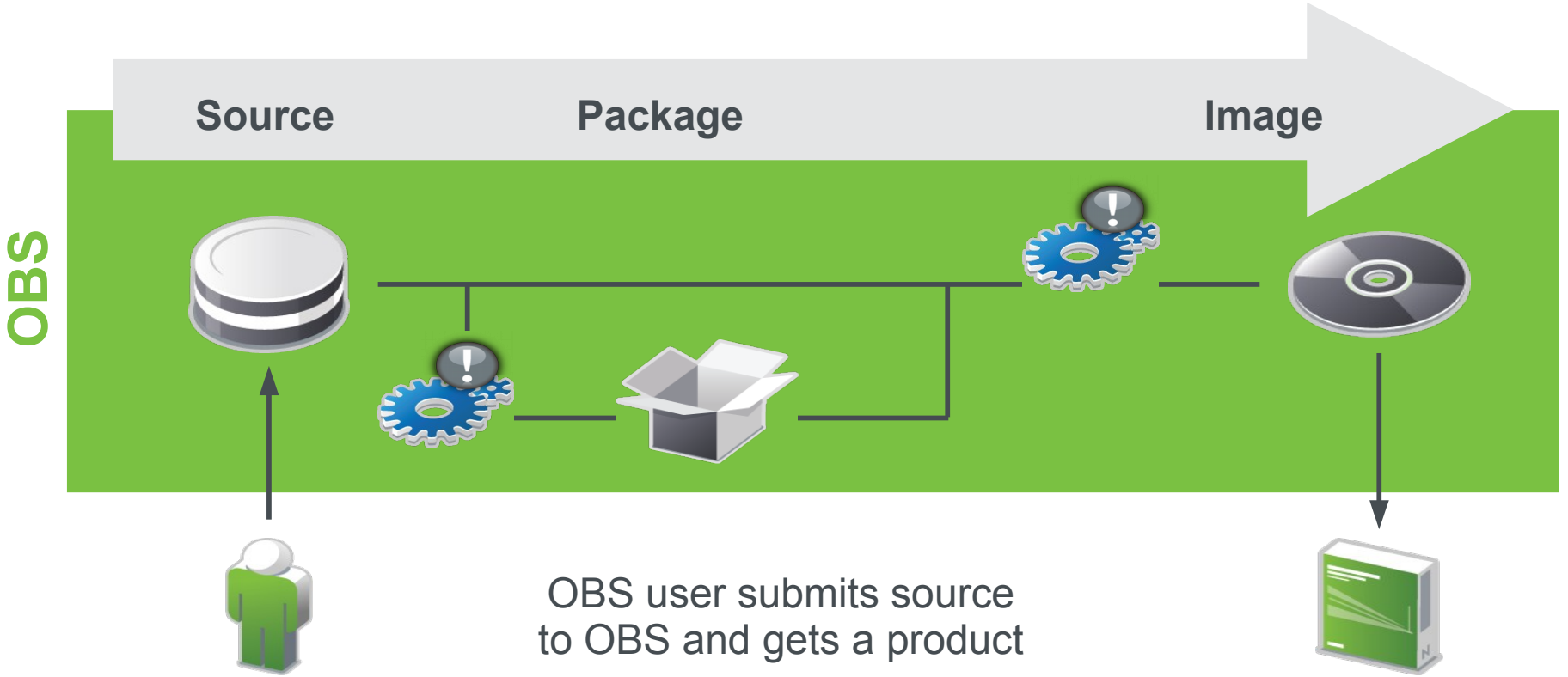
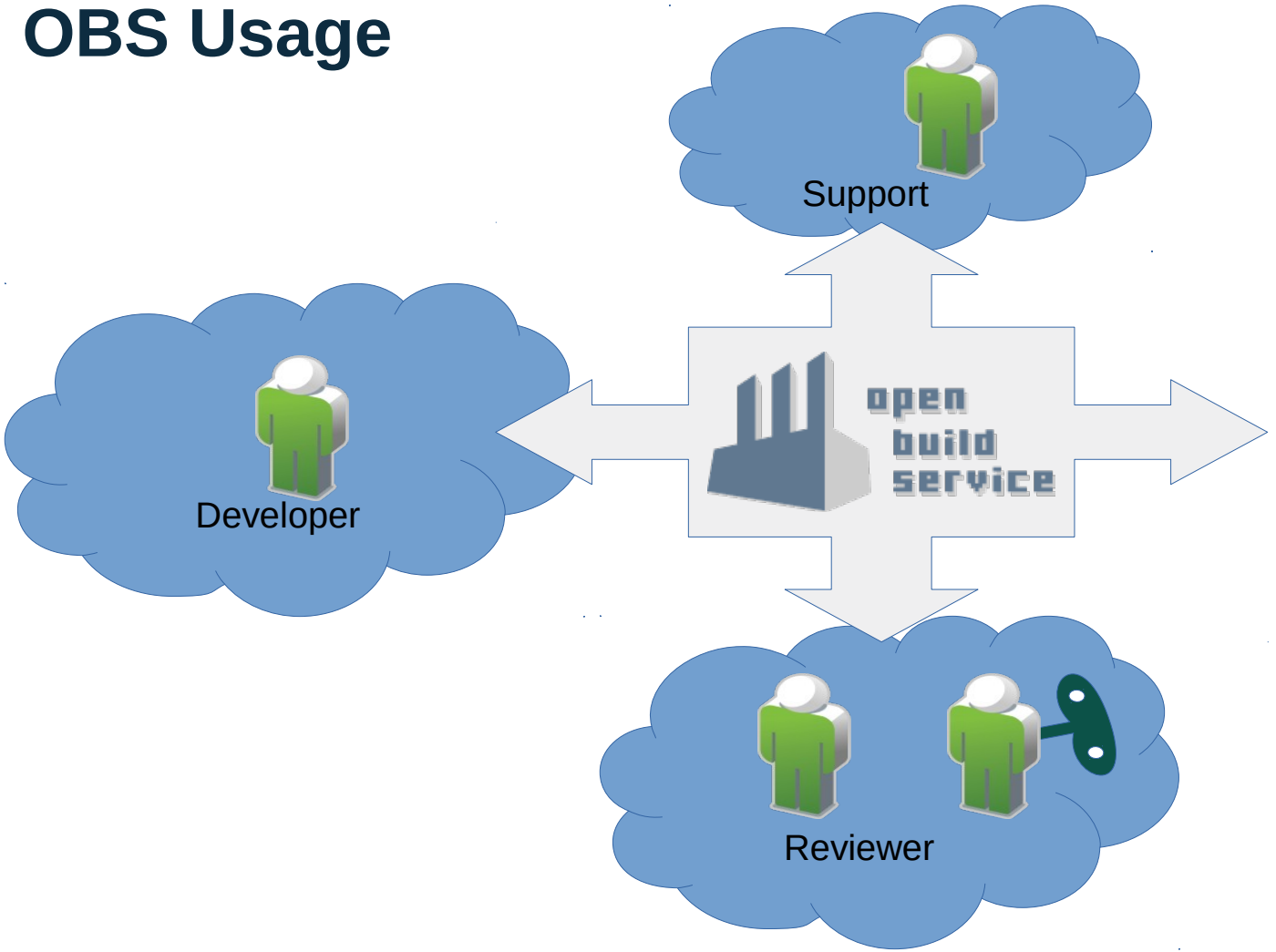
DevOps using Open Build Service

Adrian Schröter
Project Manager Build Service
<adrian@suse.de>

What is the Open Build Service(OBS)?



OBS Usage



Product

Maintenance Updates

PTF Updates

Users

- Distribution development, Maintenance Updates

- Open Source Communities

- Researchers/Universities
- Administration Teams

Support

- Community
  - opensuse-buildservice@opensuse.org
  - Irc: #opensuse-buildservice on freenode
- Professional

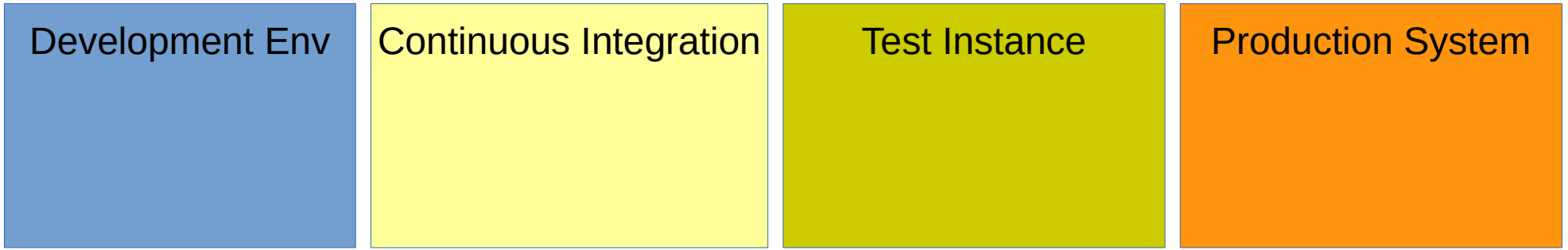
http://www.open-build-service.org/contact/

B1 Systems (L3 backing by SUSE)

DevOps

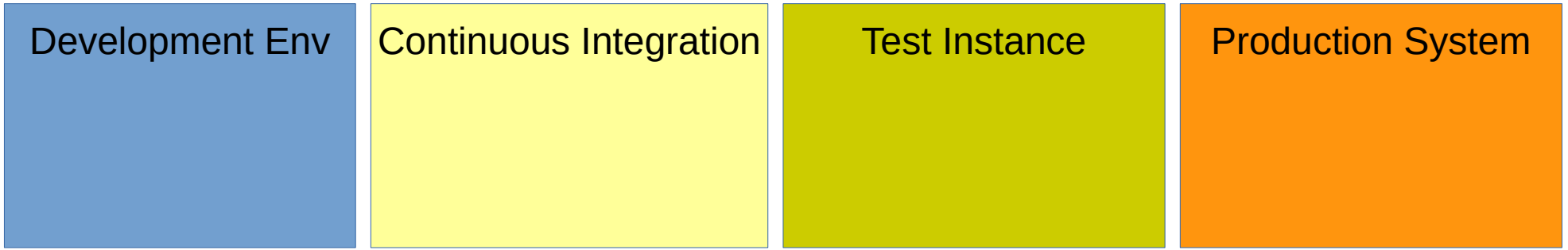
Development Pipeline

The classic DevOps setup.



Development Pipeline

The classic DevOps setup.



Workstation

- Development
- Unit tests

Working local using OBS

Getting a local copy

```
# osc checkout science:unstable FreeCAD

# cd science:unstable/FreeCAD
```

Working local using OBS

Getting a local copy

```
# osc checkout openSUSE:Factory FreeCAD
# cd openSUSE:Factory/FreeCAD
```

Running a local build

```
# osc build
```

Working local using OBS

Getting a local copy

```
# osc checkout openSUSE:Factory FreeCAD
# cd openSUSE:Factory/FreeCAD
```

Running a local build

```
# osc build
```

```
# osc build –alternate-project SUSE:SLE-12:GA
```

Testing using OBS

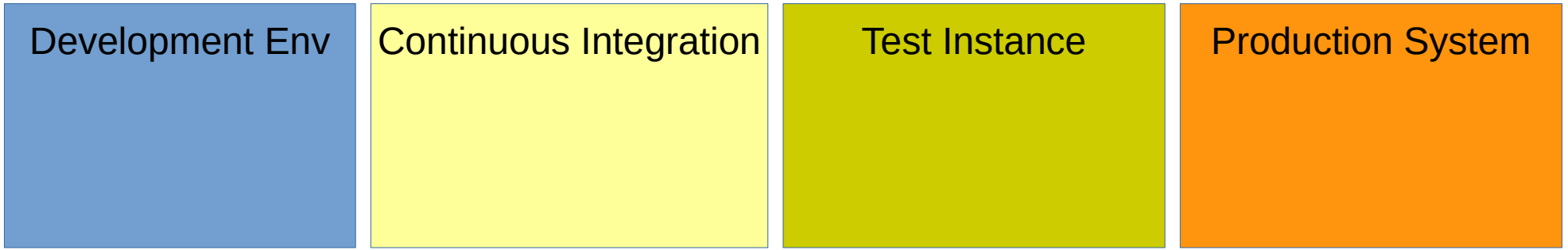
Tests are part of package build

- eg. use %check section in rpm spec files
- avoid network usage to stay reproducable

“Unit tests” are single package builds running their test-suite here.

Development Pipeline

The classic DevOps setup.



Workstation

- Development
- Unit tests

Automated Building

- Build latest code
- Functional tests

Follow a remote source

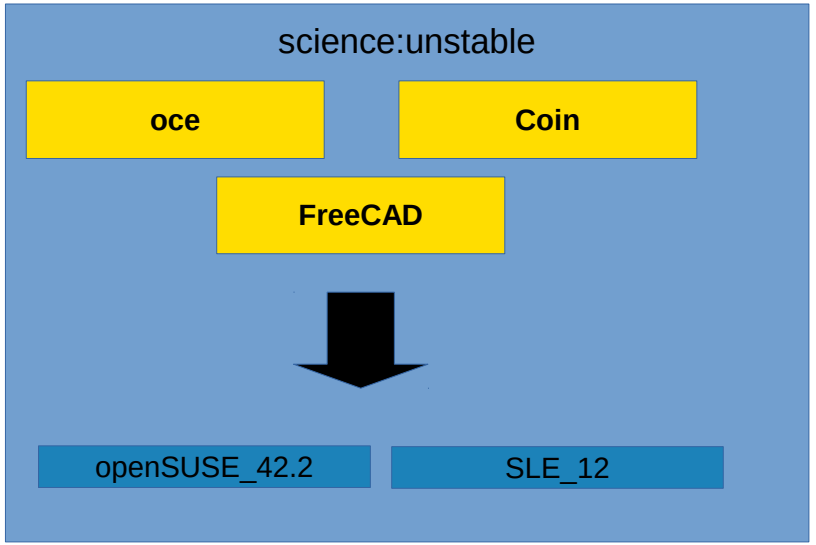
For example a git repository

 # osc add https://github.com/FreeCAD/FreeCAD.git

Creates a _service file describing how to download the source and to create a tar ball for packaging.

Testing using OBS

Packages in the same project can influence each other



Project

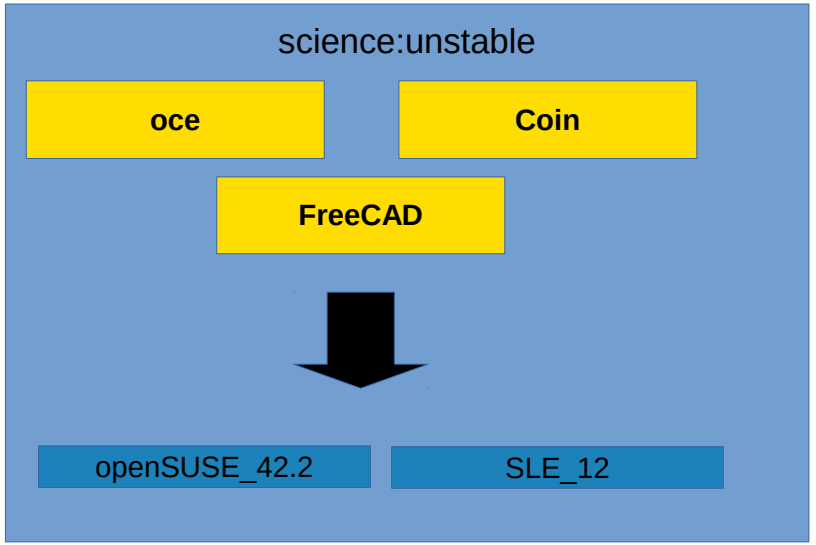
Package Container

Code Stream Repository

“Functional tests” are a set of package builds running their test-suite.

Testing using OBS

Packages in the same project can influence each other



- OBS calculates the need of rebuilding
- Consistent builds are guaranteed
- Multiple distributions from the same source

“Functional tests” are a set of package builds running their test-suite.

Create Test Instances

Creating a personal branch

 # osc branch science:unstable oce

Creates new home:$USER:branches:science:unstable project

“On demand tests” are easy possible

Create Test Instances

Creating a personal branch

```
# osc branch science:unstable oce
```

- Local changes can be applied via spec file
- The git branch can be changed via editing _service file
- Changes supposed to become part of SCM can be tested local

Create Test Instances

Check if a change affects other packages

The project can be linked against origin project to rebuild all affecting packages.

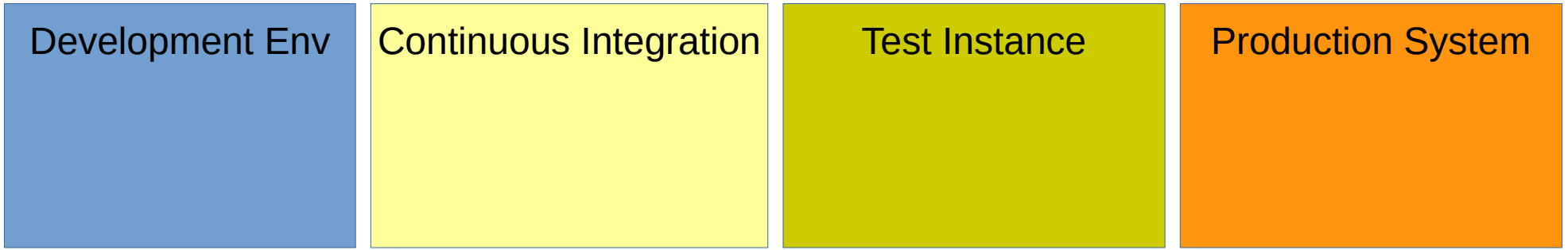
=> Check openSUSE:Factory:Staging projects for examples

OBS vs Jenkins vs Travis-ci

	OBS	Jenkins	Travis-ci
Easy to run test	- packaging needed	o scripts can be used	+ Online Available
Maintain Targets	+ Just there via interconnect	O Manual work	- Limited to one distro
Reproducability	+	-	-
Scalability	+	o	o

Development Pipeline

The classic DevOps setup.



Workstation

- Development
- Unit tests

Automated Building

- Build latest code
- Functional tests

Automated Deployment

- User testing
- Load tests

Automated deployment

Ways to deploy

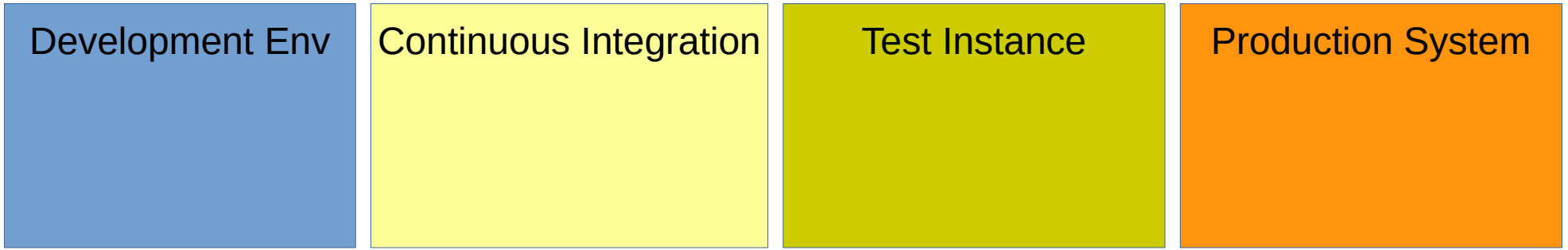
- Manual package update on test system
- Trigger update using a hook in OBS publisher
- Deploy images via PXE

Coming:

- Automatic distribution to cloud

Development Pipeline

The classic DevOps setup.



Workstation

- Development
- Unit tests

Automated Building

- Build latest code
- Functional tests

Automated Deployment

- User testing
- Load tests

- Monitoring
- Error reporting

Integrate Release Mechanism

OBS Release Mechanism

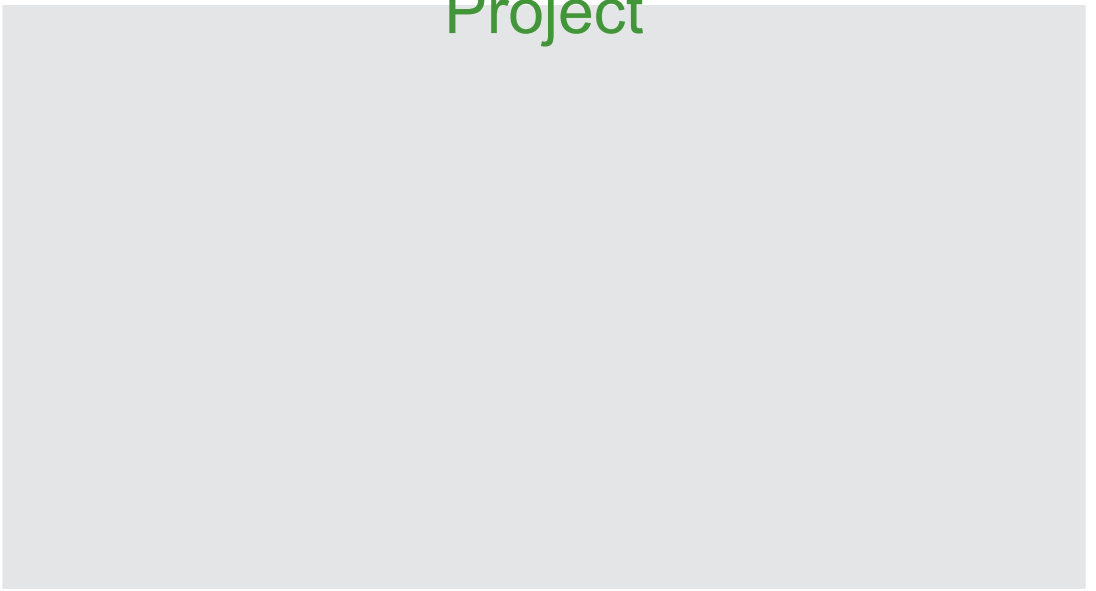
OBS release copies sources and binaries, eg:

Project:Test => Project:Stable

Building Scenarios

Own Set of Packages

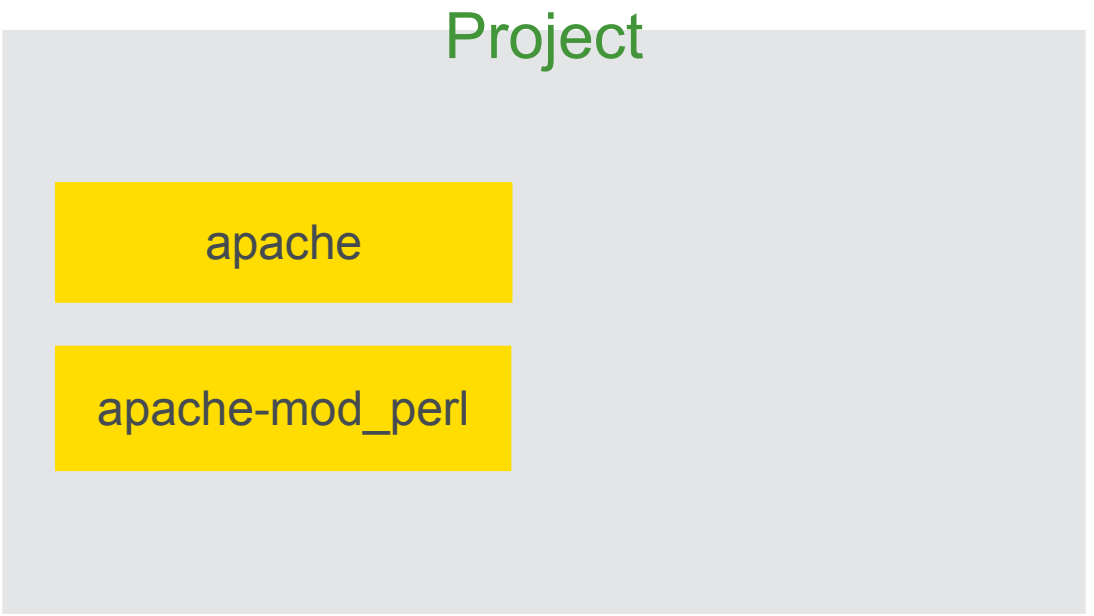
Create project



Own Set of Packages

Create project

Add package sources

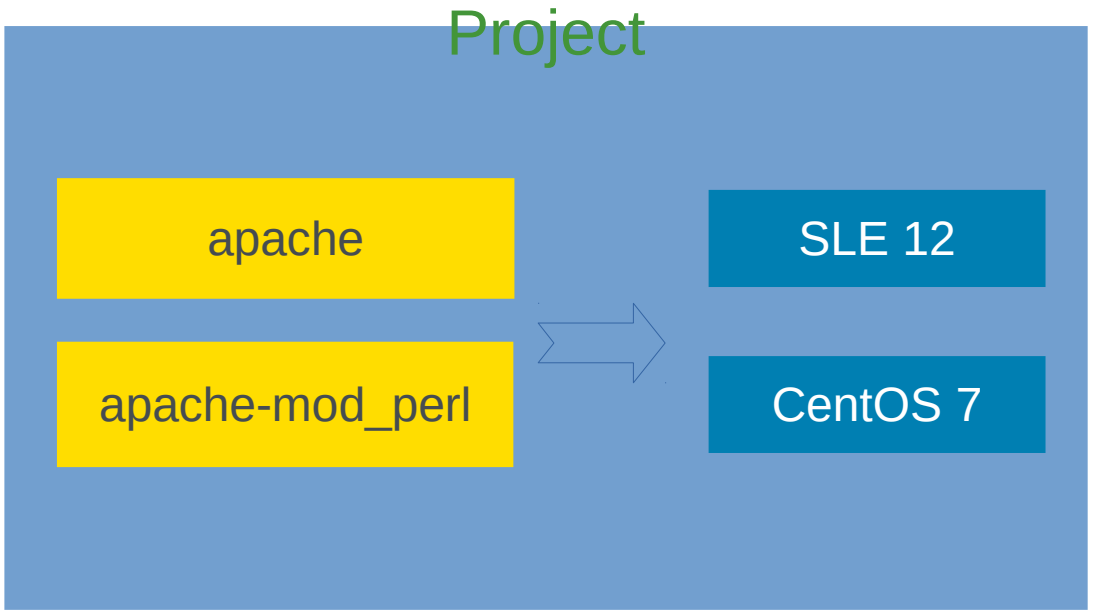


Own Set of Packages

Create project

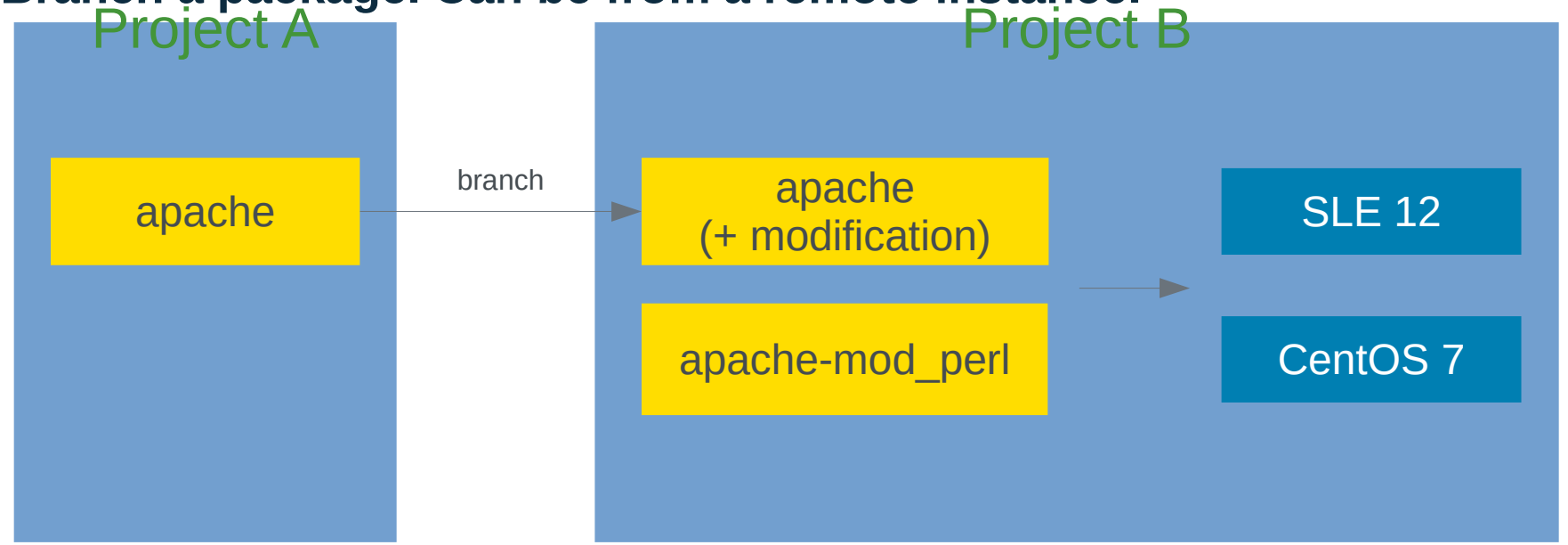
Add packages

Add targets



A Variation of a Package

Branch a package. Can be from a remote instance.



Rebuild a Part of Entire Project

Link a project

replace a package source

Configure repository to rebuild all packages affected by this package


SUSE®
We adapt. You succeed.